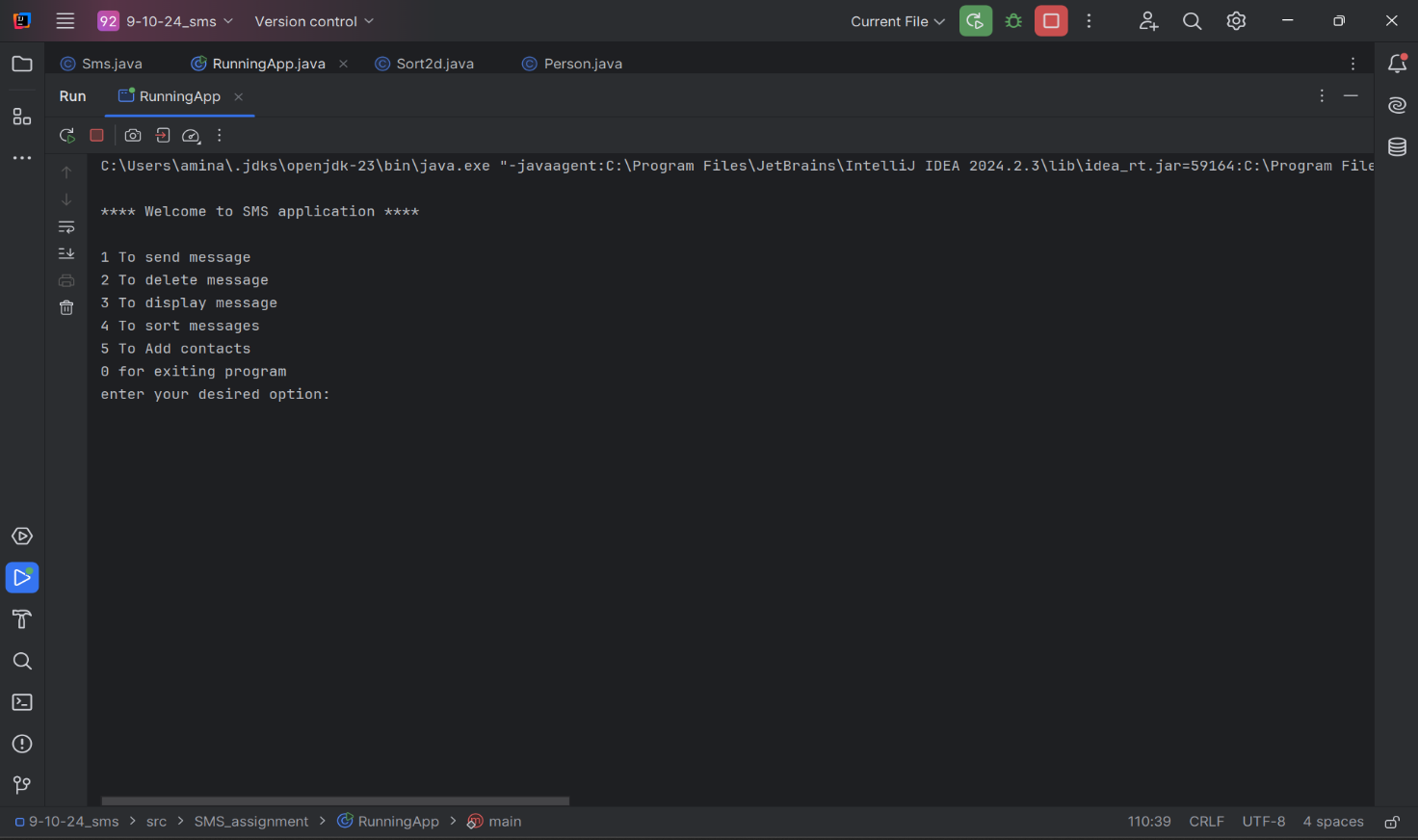
**Java Console SMS Application**

In this app a menu selection appears on the console and we have to choose our option or the function we want to run.

**Main Console:**



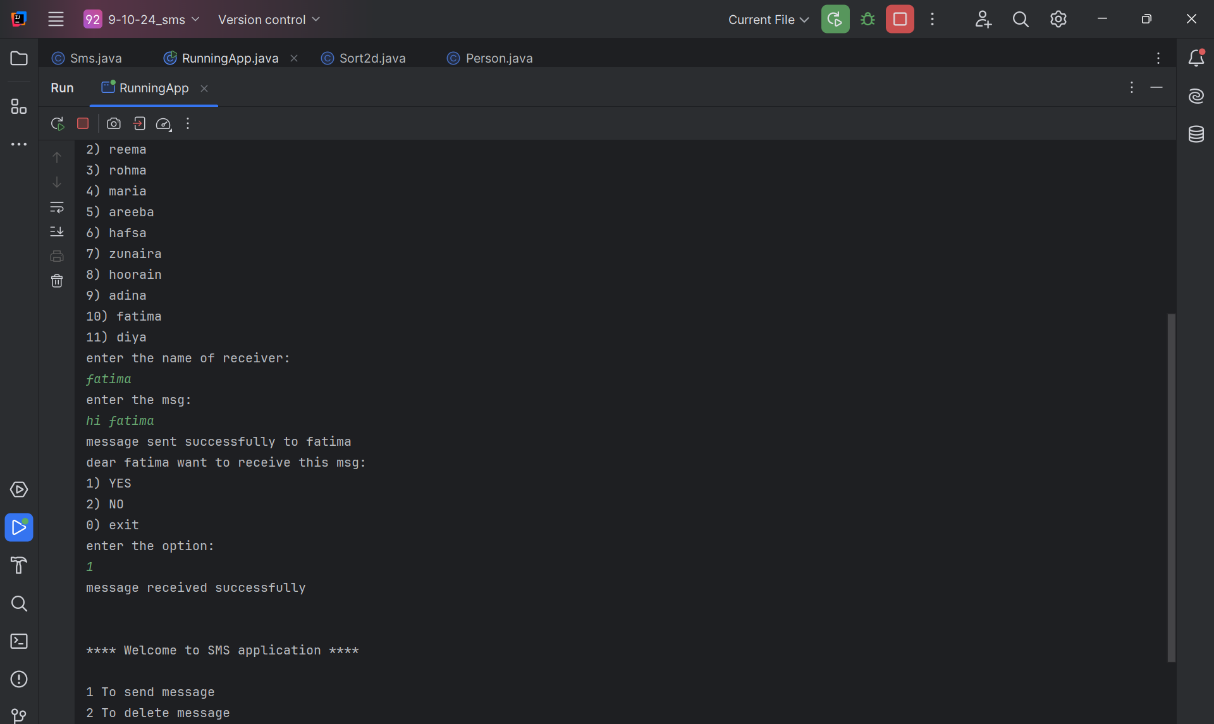
Following are the functions we can see on our screen when we run the program:

1. Send Message:
2. Delete Message
3. Display Message
4. Sort Messages
5. Add Contacts

Send message:

In this function, we are shown the list of contacts already saved in our array we can choose the contact to whom we want to send message then we write the message to be sent. After sending the messages it asks us if we want to receive the message we just sent we choose our option and if message is received by us it has a time stored at which we received the message.

Then initial menu of 5 primitive functions reappear and we can consider any of those options again.

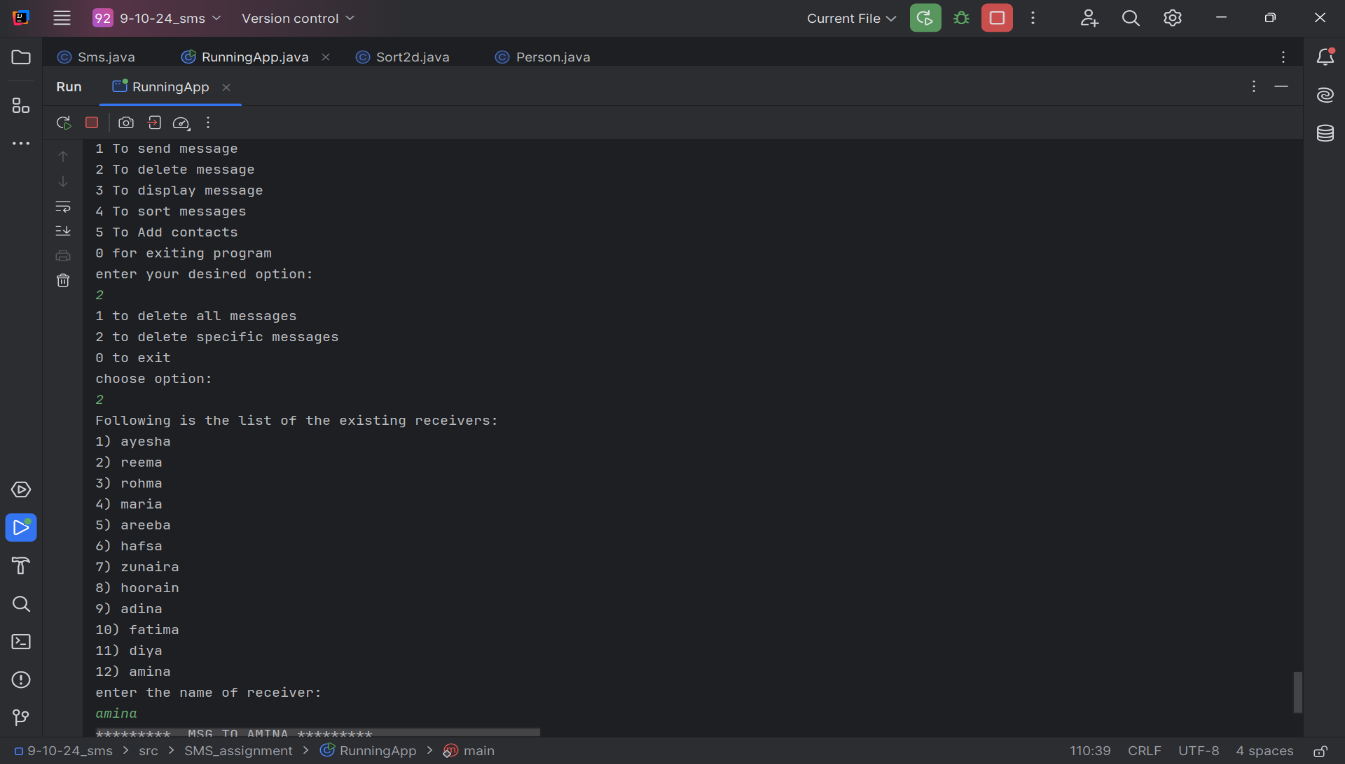


Delete message:

When we choose the delete function we are shown 2 more options to choose from we can either delete all messages or we can delete any specific message:

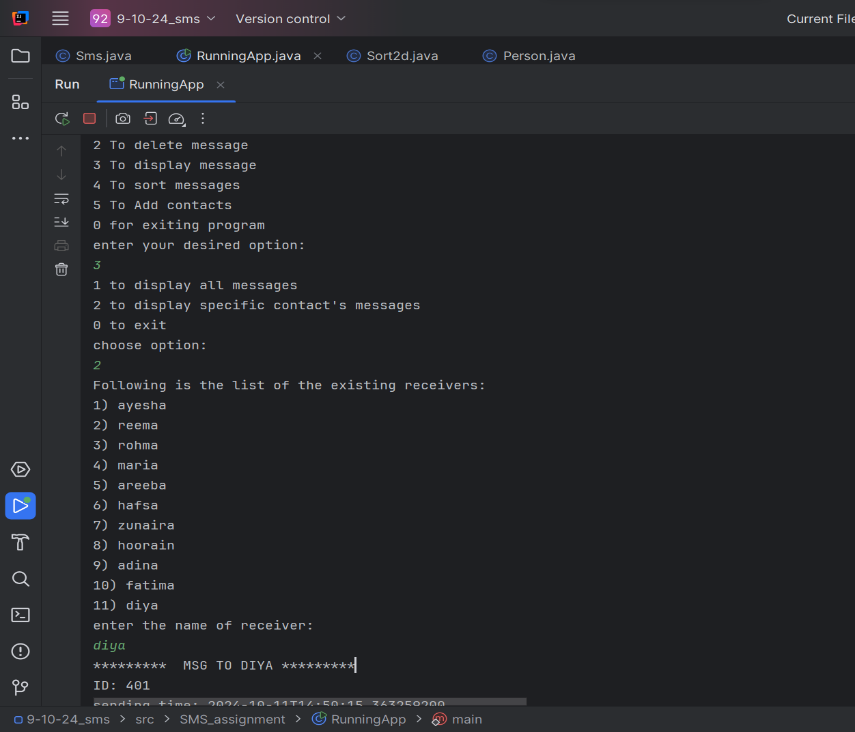
1. Delete all messages
2. Delete specific message

if we choose to delete any specific message than we first choose from the saved contacts it displays all the existing messages of that contact and ask for the input of the ID of the message we want to choose from all those which are displayed. Deleting here means considering it null.



Display message:

This also gives us further more options to choose from it asks us if we want to display all the messages of all the existing contacts or maybe just messages sent to a single contact:

1. Display all messages
2. Display messages of a specific

contact

Displaying all messages in an array is

quite easy but for second option i

took the input if the contact whose

messages user want to display and

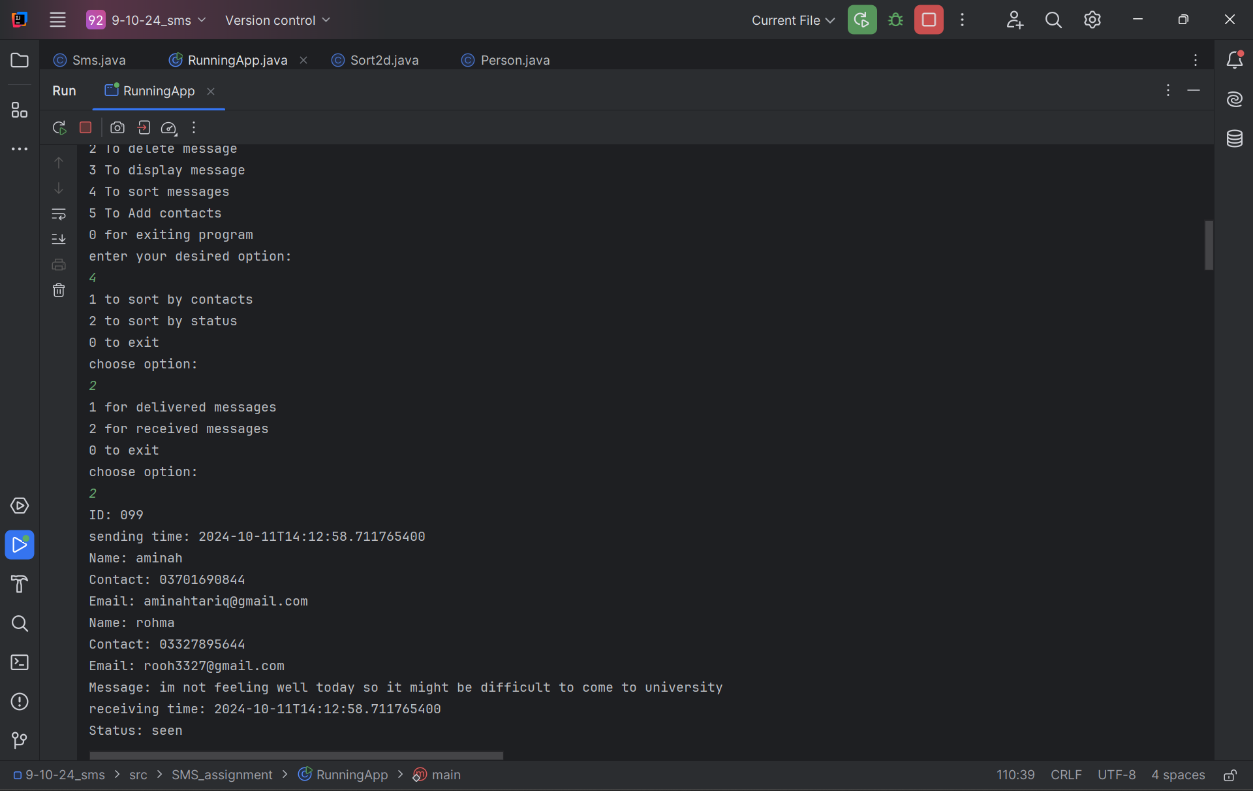
find it using .equals() hence

displaying the messages.

Sort messages:

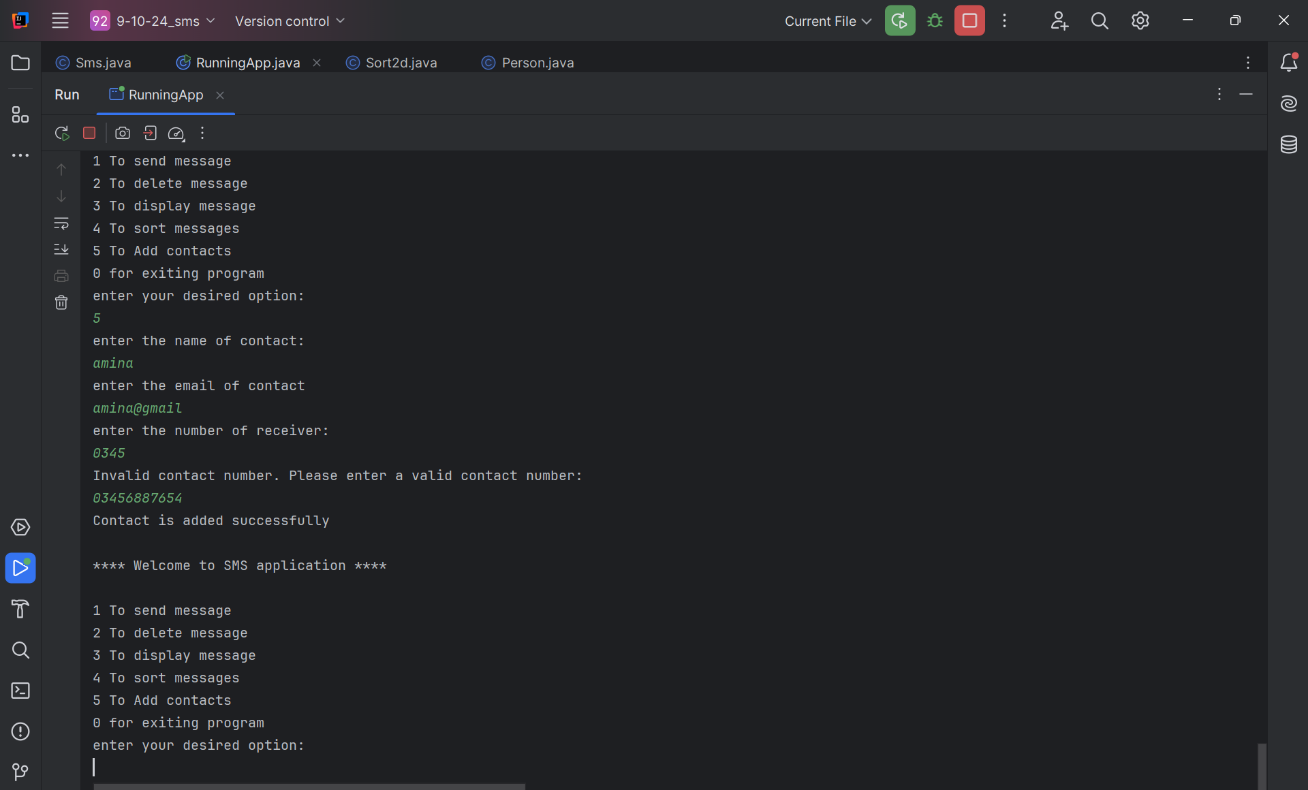
To sort messages we can either do it by contacts or by their status if the message has been received or not

In contacts it is again option if the specific contact’s messages are to be displayed all of them or they should again be sorted by the received r delivered category



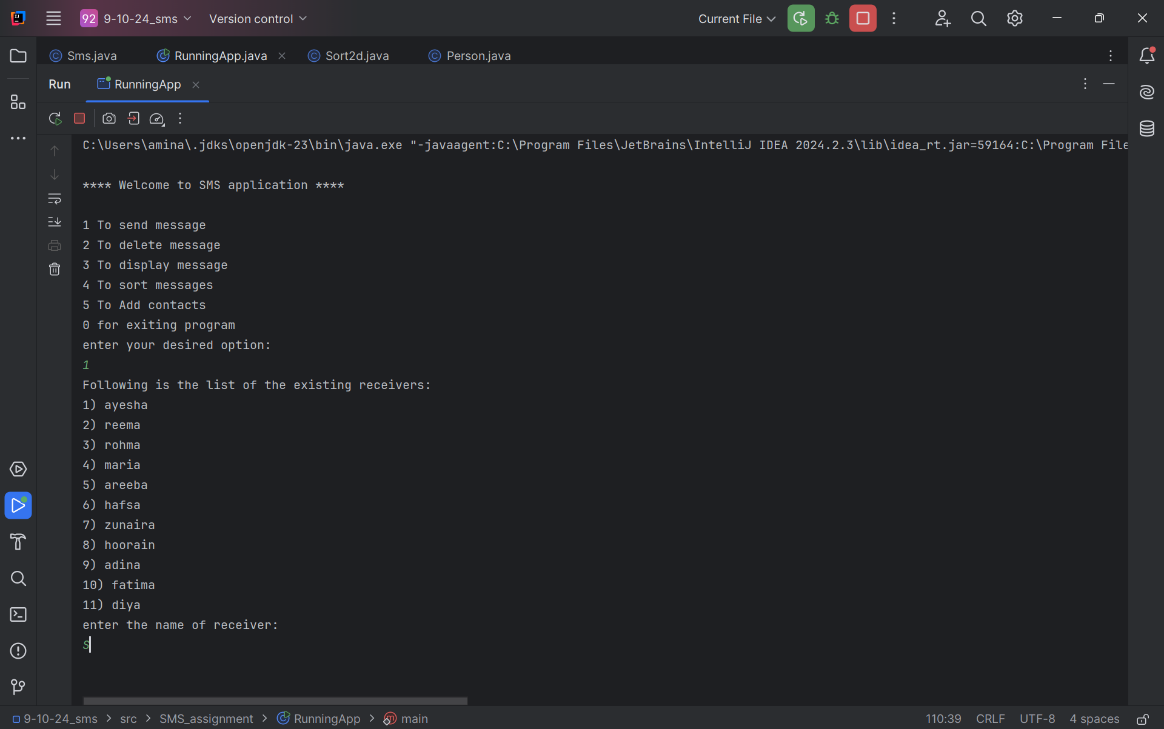
Add contact:

If we want to add another contact in our array we can use the last function it takes the name, email and the phone number of the receiver to store in the array so next time when we want to choose from contacts we can have new contacts as our options



**Additional points:**

Now considering all these function are run time environment that means any changes in the array done after its initialization will be no more once we exit the program.

The 2d array is considered as rum time database here.

Repition of code

Now some of the code was used in

repetitions so i made more function

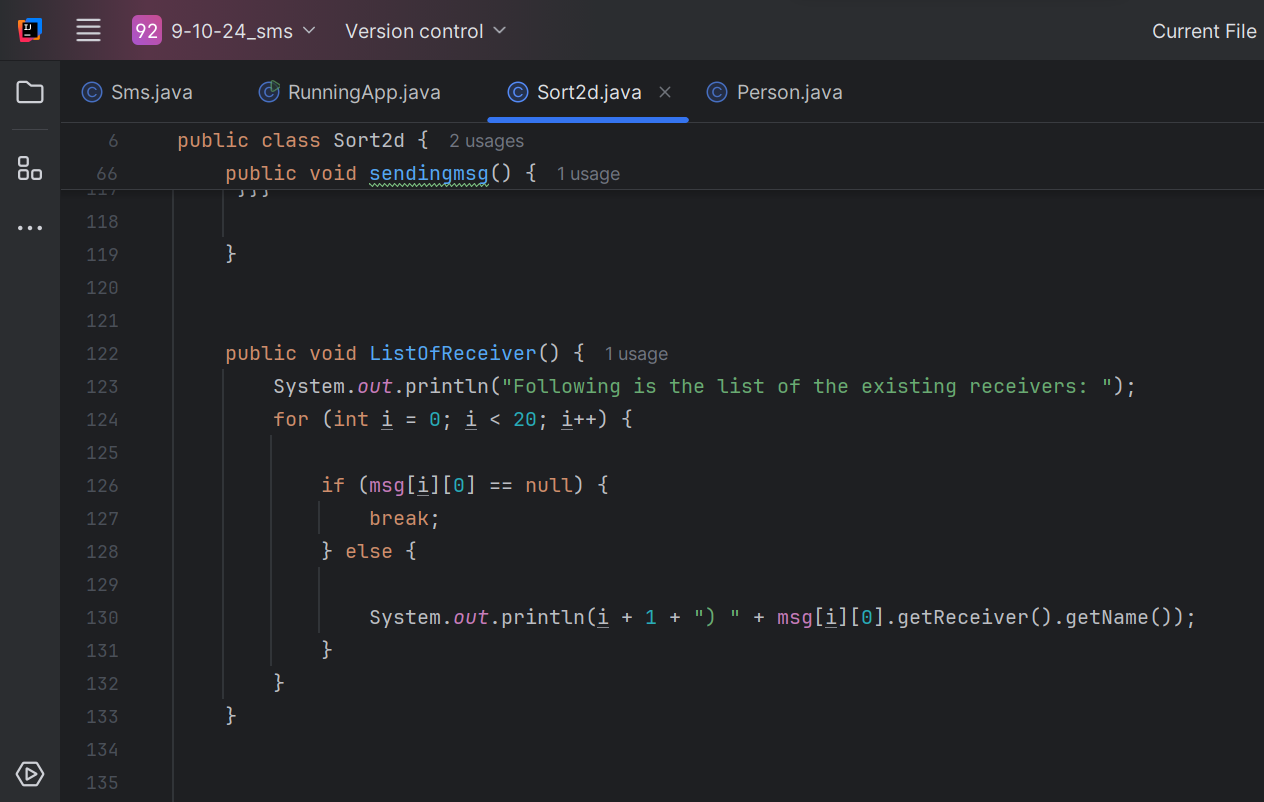
like the input or choice from the

list of the contacts and verify if the

contact exists inour list or not hence

whenever user give invalid input it

asks for the input again.



Same goes with the contact number of the receiver as it has 11 characters which is a must format hence user cannot proceed unti and unless it give valid input

The main console is repeated again again so it is also a function making it easy for my to debug and read my code

Other attributes:

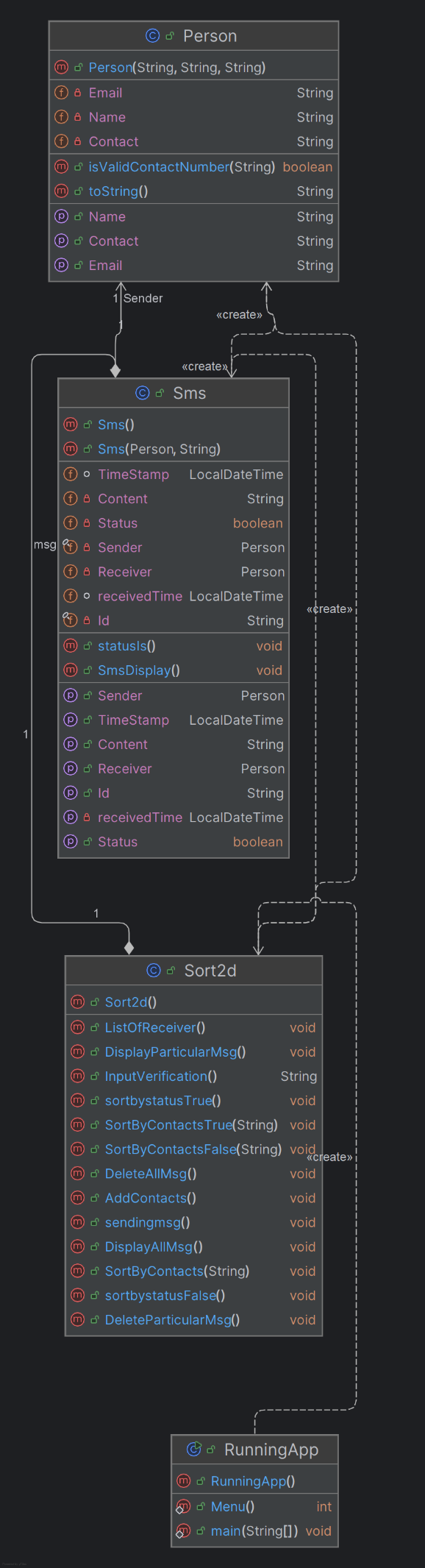
Sender in my application is final variable as it is only the user who sends messages to different contacts whenever the messages is received it also saves the exact time the message is received and it cannot be changed unseen messages or unreceived messages do not have receiving time

My constructor of 2d array class initializes 11 contacts having 40 sms each and i have randomly received some of the messages for the sake of the project.

**UML diagram:**

This diagram shows :

1. Classes used in the project
2. Their attributes
3. Their methods
4. Their dependencies
5. Their constructors



Hence this is the structure of whole application